

Pre-Trained Denoising Autoencoders Long Short-Term Memory Networks as probabilistic Models for Estimation of Distribution Genetic Programming

Kolloquium zur Masterarbeit im M.Sc. Wirtschaftspädagogik

Roman Hoehn

JGU Mainz

Datum: 25.01.2023

- 1 Einleitung**
- 2 Denoising Autoencoder Genetic Programming**
- 3 Aktueller Forschungsstand**
- 4 Implementation**
- 5 Untersuchung des Suchverhaltens bei symbolischer Regression**
- 6 Alternative Pre-Training Strategien**
- 7 Fazit (vorläufig)**

Section 1

Einleitung

Forschungsfrage

Kann das Suchverhalten der Denoising Autoencoder Genetic Programming (DAE-GP) Metaheuristik durch den Einsatz einer Pre-Training Strategie optimiert werden?

Welchen Effekt hat Pre-Training auf:

- ① die Qualität der gefundenen Programme (Fitness/Programmlänge)?
- ② die Populationsdiversität?
- ③ das Laufzeitverhalten?

Anwendungsgebiet: Symbolische Regression, Fokus auf Airfoil Datensatz

Section 2

Denoising Autoencoder Genetic Programming

Übersicht

- Metaheuristik basierend auf genetischer Programmierung (GP)
- Ersetzung der Variationsoperatoren von GP durch künstliche, neuronale Netze zur Optimierung des Suchverhaltens¹
- Variante des Estimation of Distribution-GP (EDA-GP)
- Einsatz von Pre-Training in mehreren Publikationen als möglicher Weg für eine weitere Optimierung vorgeschlagen² ³

¹Wittenberg, Rothlauf und Schweim (2020)

²Wittenberg und Rothlauf (2022)

³Wittenberg (2022)

Estimation of Distribution Algorithmen⁴ ⁵ (EDA)

- Entwicklung neuer Rekombinationsoperatoren für evolutionäre Algorithmen basierend auf dem Einsatz von probabilistischen Modellen
- Hypothese: Problemspezifische Abhängigkeiten zwischen Entscheidungsvariablen können bei der Erzeugung neuer Individuen besser berücksichtigt werden als bei traditionellen Rekombinationsoperatoren

⁴Mühlenbein und Paass (1996)

⁵Rothlauf (2011)

Denoising Autoencoder Estimation of Distribution Algorithmen (DAE-EDA)

Idee: Einsatz von Denoising Autoencoders⁶ (DAE) als probabilistisches Modell für genetische Algorithmen⁷

2 Phasen Ansatz:

- ① Model Building: Modell “lernt” die Eigenschaften von ausgewählten Lösungen hoher Güte durch das Trainieren eines DAE
- ② Model Sampling: Neue Lösungen werden erzeugt durch das propagieren von bestehenden, mutierten Lösungen durch das erlernte Modell

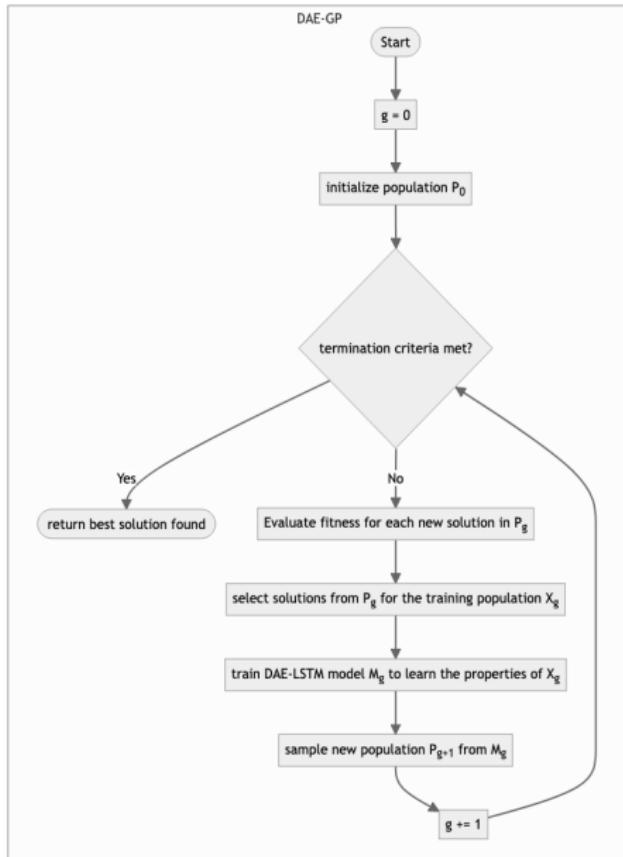
⁶Vincent u. a. (2008)

⁷Probst und Rothlauf (2020)

Denoising Autoencoder Genetic Programming (DAE-GP)

- Adaptierung des DAE-EDA Algorithmus auf GP
- Schwierigkeit: Variation von GP Lösungen in Länge und Baumstruktur
- seq2seq learning Problem: Einsatz von DAE - Long Short Term Memory Netzwerken (DAE-LSTM)

DAE-GP Ablauf



Pre-Training

Idee: Modelle werden vor ihrem eigentlichen Einsatz zum Lösen eines Problems auf möglichst großen Datensätzen vorgenommen

Mögliche Vorteile durch Pre-Training⁸:

- ① Geringere Bedarf an Trainings Daten für vorgenommene Modelle
- ② Reduktion der Trainingszeiten/Laufzeiten
- ③ Verbesserung der Güte des Modells
- ④ Verbessertes Generalisierungsverhalten des Modells

⁸Erhan u. a. (2009)

Section 3

Aktueller Forschungsstand

Generalisiertes Royal Tree Problem⁹ (GRT):

- Einfaches Suchproblem mit hoher Lokalität
- DAE-GP erzeugt durch Model Sampling Lösungskandidaten mit höherer Fitness als GP
- Hohe Güte der erzeugten Lösungskandidaten resultiert in besserer Performance
- Perfomance Vorteil steigt mit zunehmender Komplexität des GRT Problems

⁹Wittenberg, Rothlauf und Schweim (2020)

Symbolische Regression¹⁰:

- Airfoil Datensatz für Real-World symbolische Regression
- DAE-GP erzeugt für eine vorgegebene Anzahl von Fitness Evaluationen im Vergleich zu GP:
 - ① Lösungen mit höherer Fitness
 - ② Lösungen mit geringerer Größe

¹⁰Wittenberg und Rothlauf (2022)

Pre-Training für Denoising Autoencoders¹¹

Positive Wirkung von Pre-Training auf DAE

- ① Gesteigerte Modell Performance (sinkender Testfehler)
- ② Besserer Generalisierungsfähigkeit
- ③ Erhöhter Robustness des Algorithmus (sinkende Varianz des Testfehlers)

Einfluss der Modell Architektur

- Positiver Effekt steigt mit zunehmender Komplexität des DAE
- Je mehr versteckte Layer oder Neuronen pro verstecktem Layer vorhanden sind, desto besserer Effekt des Pre-Trainings
- Für sehr kleine DAE, zeigt Pre-Training jedoch inverse, negative Auswirkung auf die Modell Performance

¹¹Erhan u. a. (2009)

Section 4

Implementation

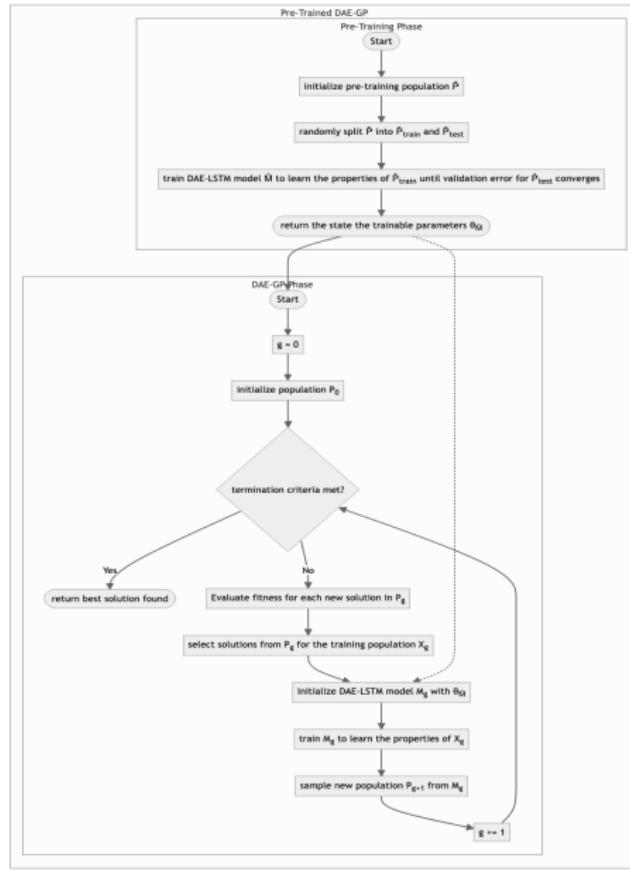
Überblick

Gewählte Pre-Training Strategie:

- (Klassisches) Pre-Training: Einmaliges Pre-Training eines DAE-LSTM mit einer großen Population aus Lösungskandidaten \hat{P}_{train} (ramped Half and Half Initialisierung)
- Trainingsmethode Early Stopping: Abbruch des Trainings sobald der Testfehler für eine separate Population \hat{P}_{test} konvergiert

Ausschluss weiterer Pre-Training Strategien wie Re-Use Learning, Few-Shot Learning

Pre-Trained DAE-GP Ablauf



Herausforderung - DAE-LSTM Dimension

- Im Verlaufe des DAE-GP Algorithmus passt sich die Dimension der DAE-LSTM Netzwerke M_g dynamisch an die Größe der Individuen der aktuellen Population an (Dimension nimmt i.d.R. stark ab innerhalb der ersten Generationen)

Der Einsatz der Pre-Training Strategie ist nur möglich bei einer konstanten Anzahl an Neuronen. Zwei Ansätze verfolgt:

- ① DAE-GP mit dynamischer Anpassung, Pre-Trained DAE-GP mit konstanter Anzahl (maximale Länge innerhalb von \hat{P})
- ② Beide Algorithmen mit einer fixen Anzahl von versteckten Neuronen

Section 5

Untersuchung des Suchverhaltens bei symbolischer Regression

Fragestellung

Welchen Einfluss hat die Verwendung einer Pre-Training Strategie auf das Suchverhalten von DAE-GP bei der Anwendung auf symbolische Regressionsprobleme?

Aufbau: Betrachtung des Suchverhalten über je 10 Gesamtdurchläufe am Airfoil Datensatz¹² für DAE-GP und pre-trained DAE-GP

Fokus insbesondere auf:

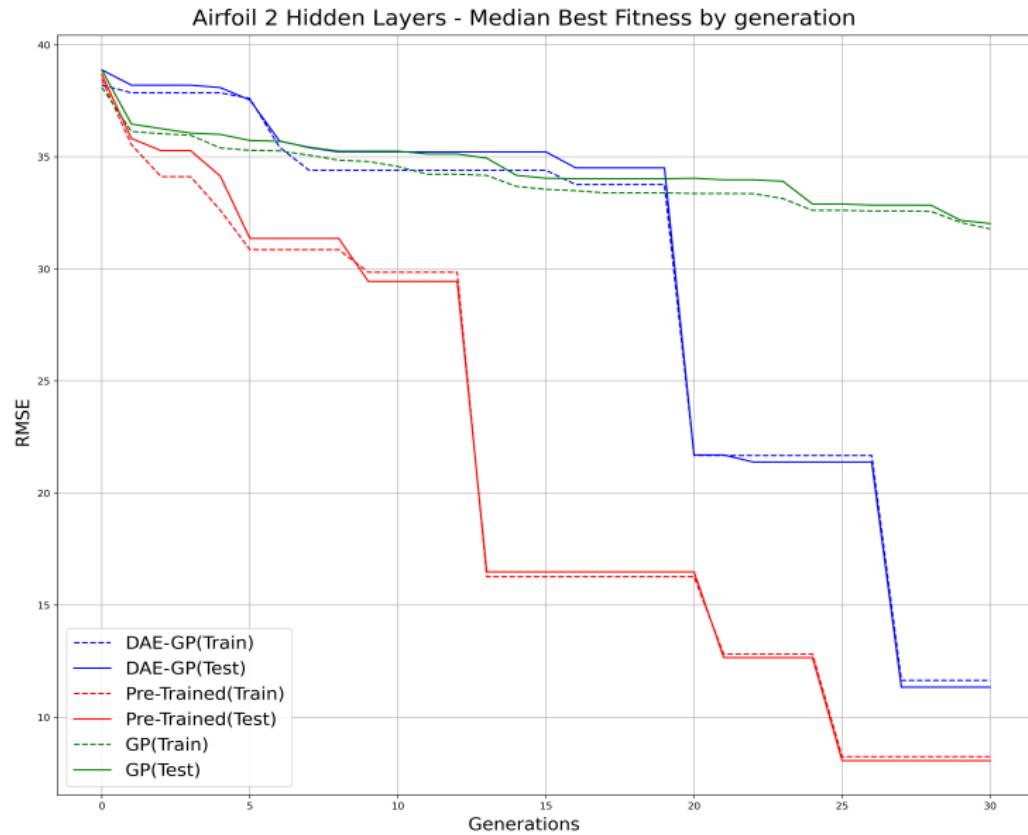
- Lösungsqualität (Fitness)
- Größe der gefundenen Lösungen (Anzahl an Knoten)
- Populationsdiversität
- Laufzeit

¹²Dua und Graff (2017)

Hyperparameter

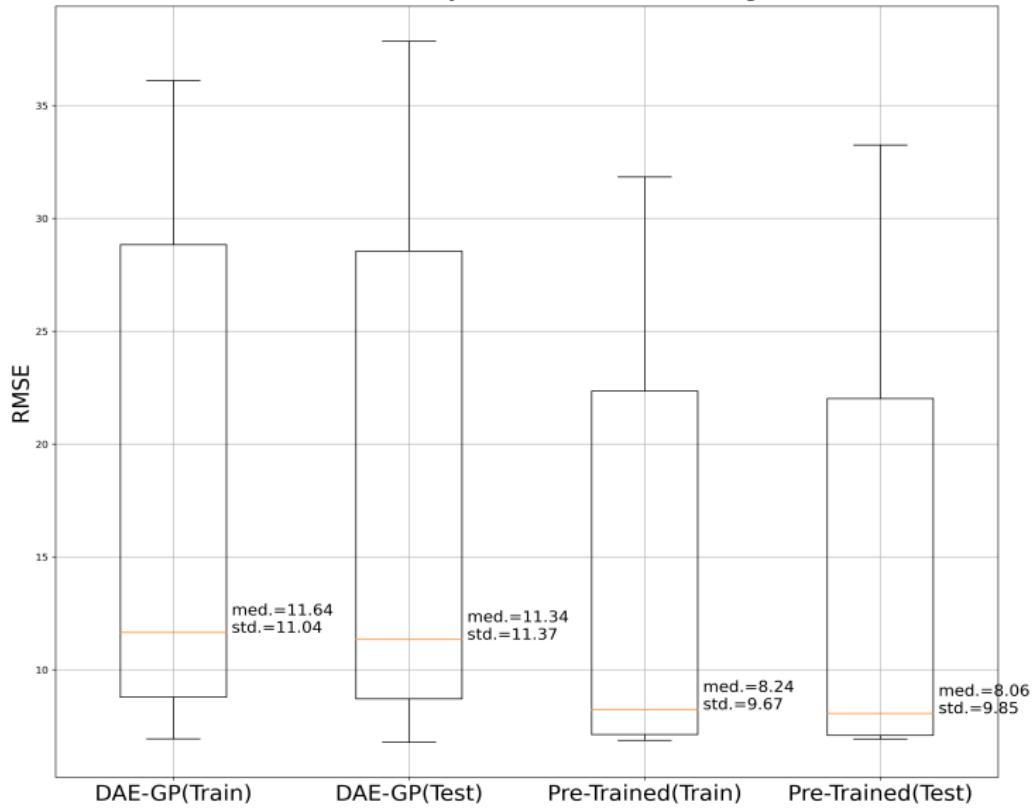
populationSize	X500
generations	30
fitness	RMSE
TrainingMode	Convergence
SamplingSteps	2
hiddenLayers	2
Selection	Binary Tournament Selection
corruptionTechnique	Levenshtein Edit
edit_training	5%
edit_sampling	95%
functionSet	{+ ; - ; * ; aq}
Pre-Training PopulationSize	10000
Pre-Training Train/Test Split	50%
Pre-Training TrainingMode	Early Stopping

Airfoil - Entwicklung der Fitness über Generationen



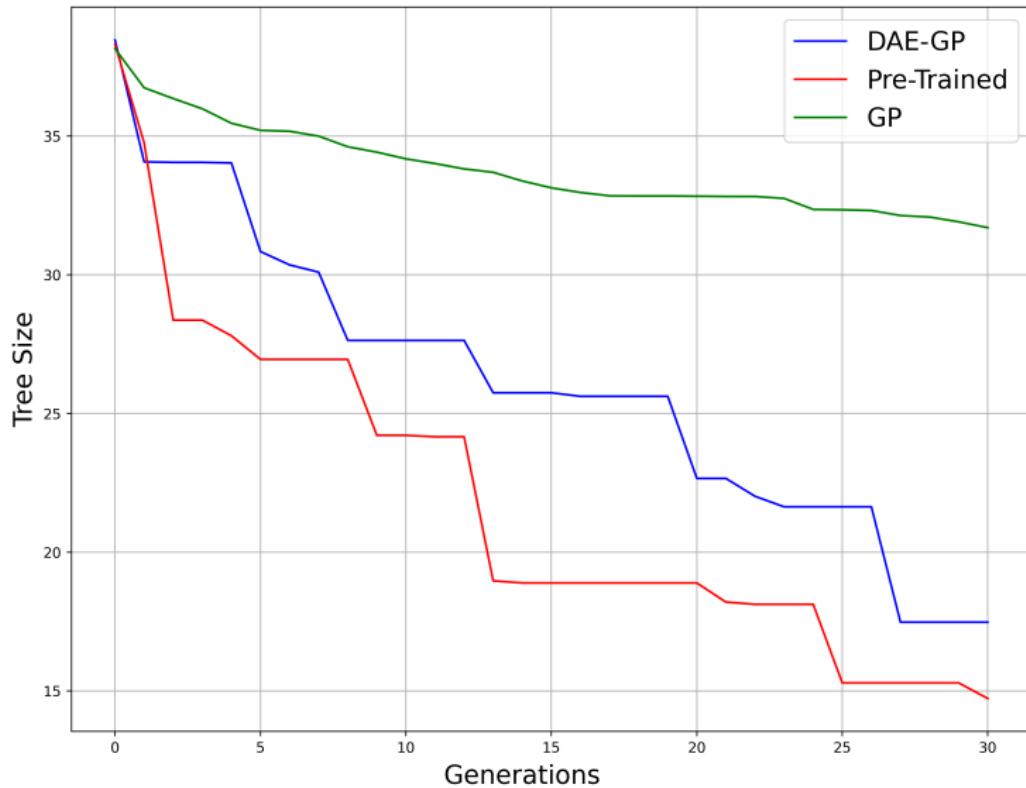
Airfoil - Verteilung der finalen Fitness

Airfoil 2 Hidden Layer - Best Fitness after 30 gens



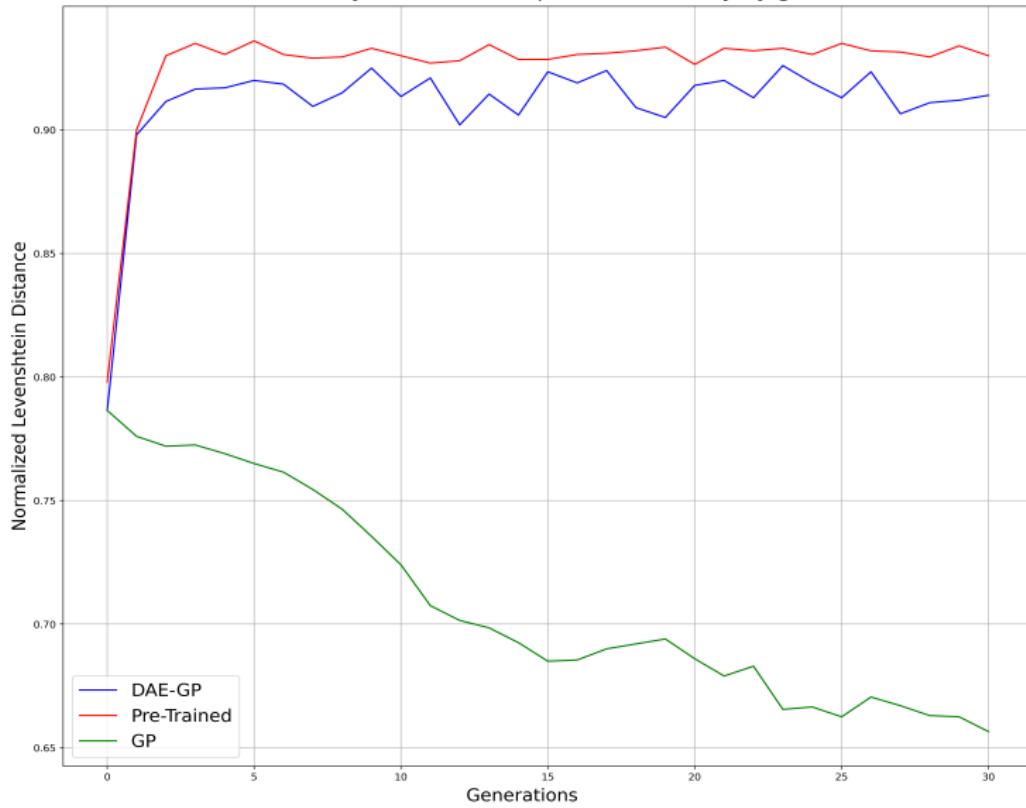
Airfoil - Größe der besten Lösungskandidaten

Airfoil 2 Hidden Layers -Median Size best Solution

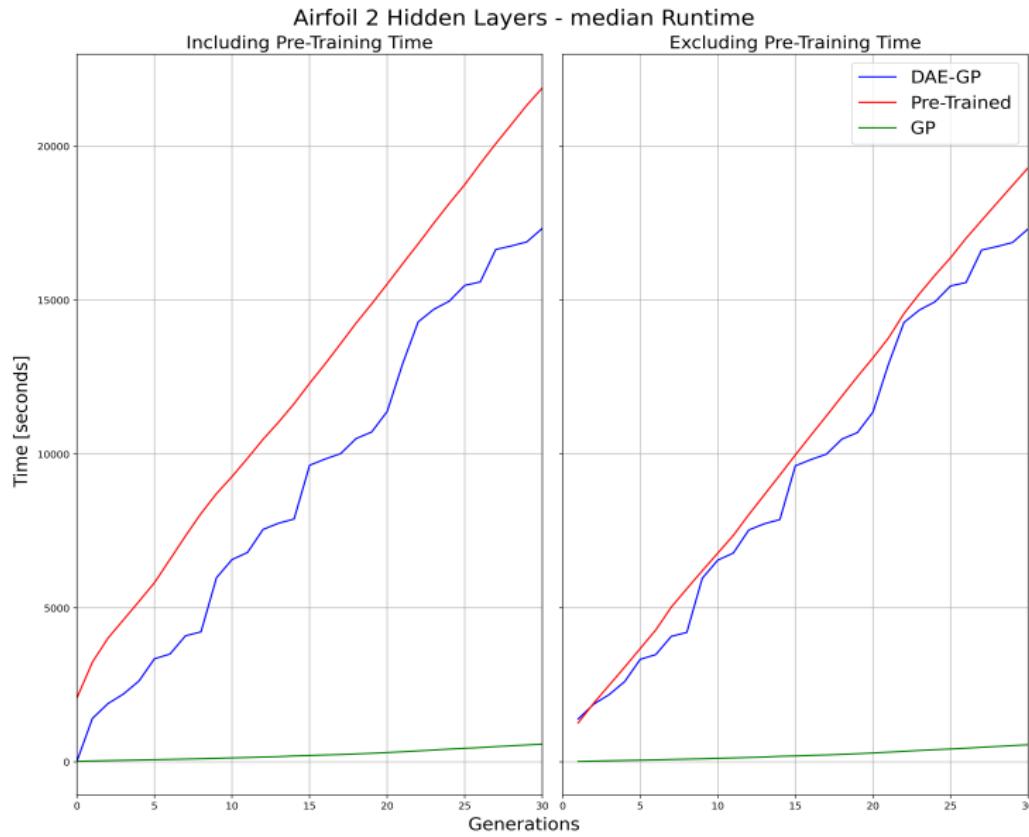


Airfoil - Populationsdiversität

Airfoil 2 Hidden Layers - Median Population Diversity by generation

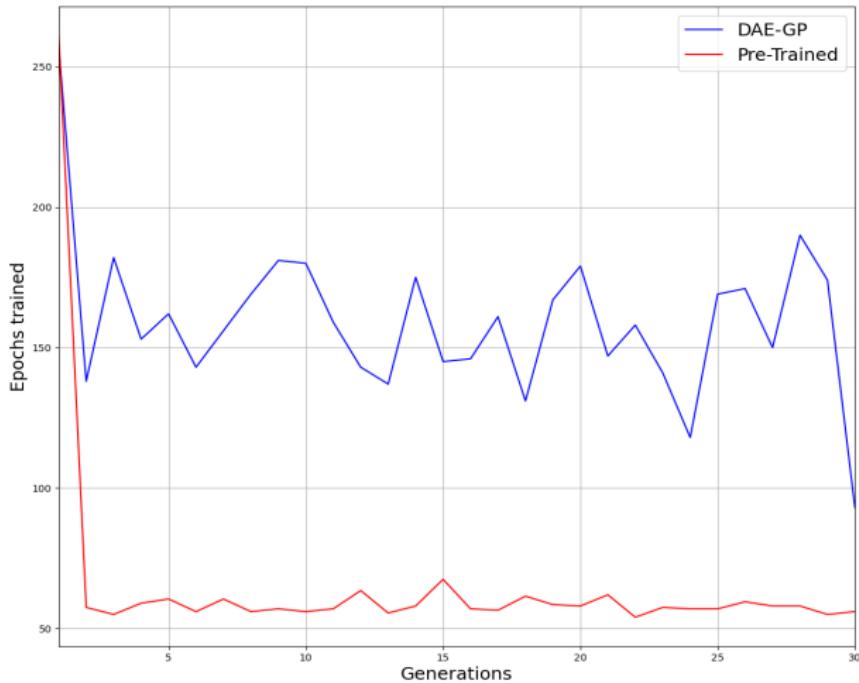


Airfoil - Laufzeit (Gesamt)

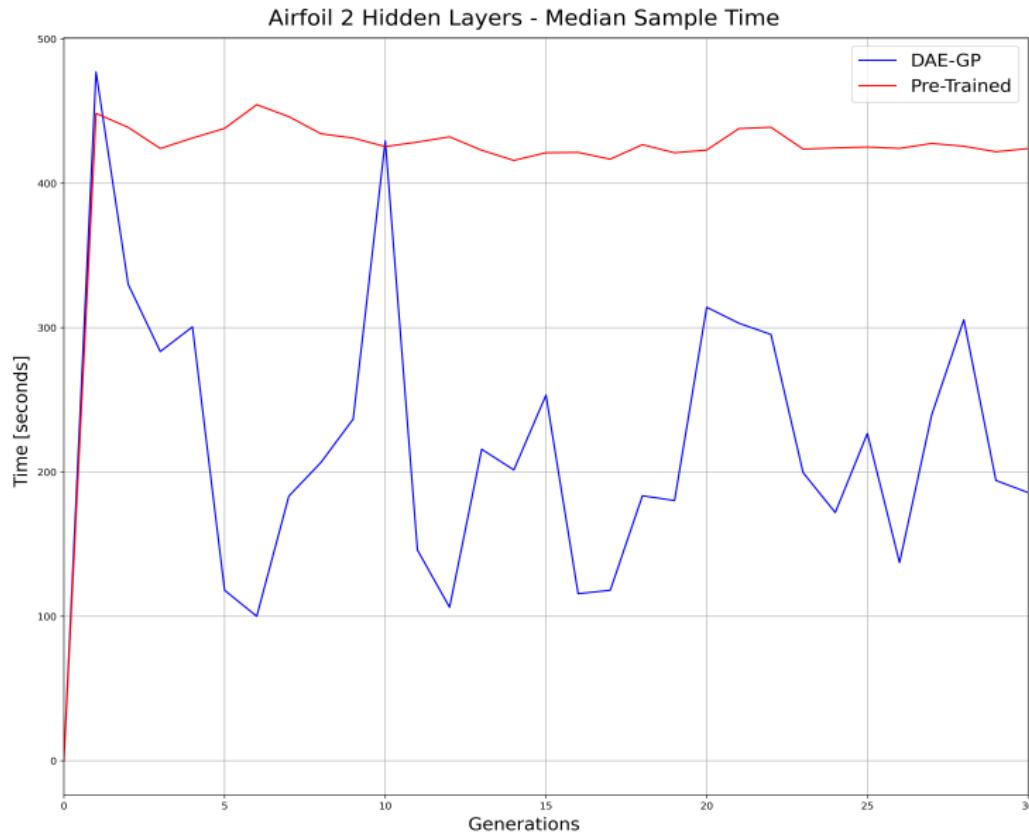


Airfoil - Laufzeit (Trainingsepochen)

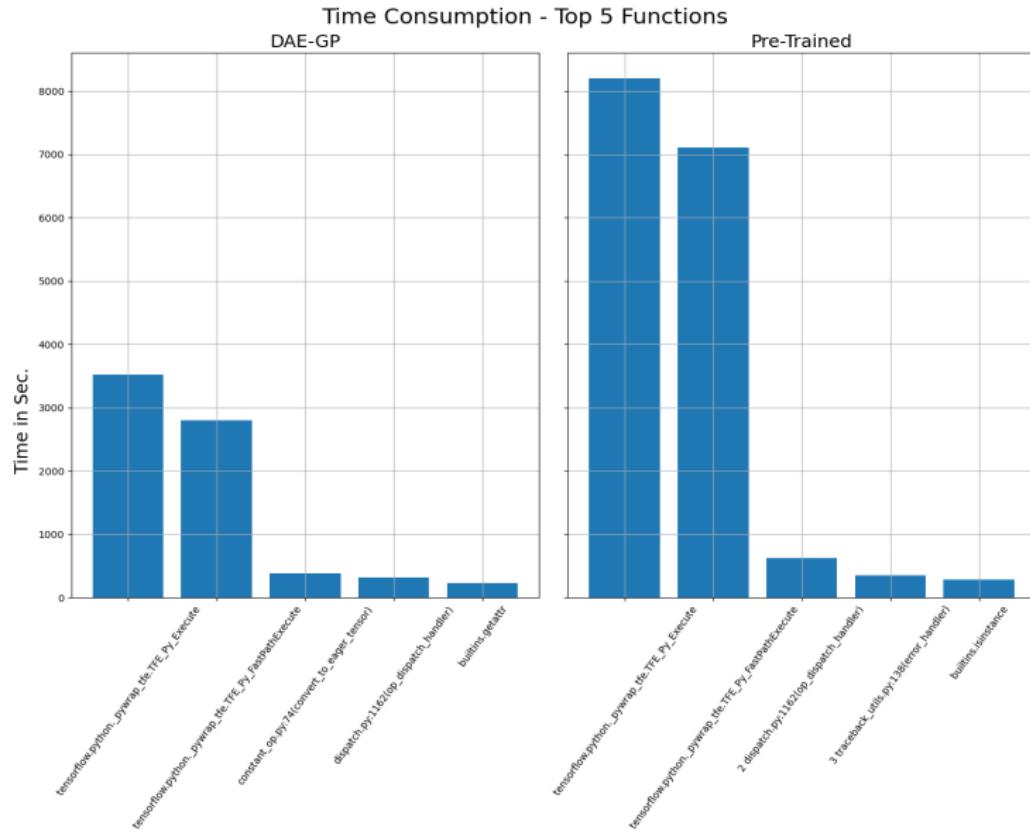
Airfoil 2 Hidden Layers - Median Training epochs



Airfoil - Laufzeit (Sampling Time)



Airfoil - Laufzeit (Profiling)



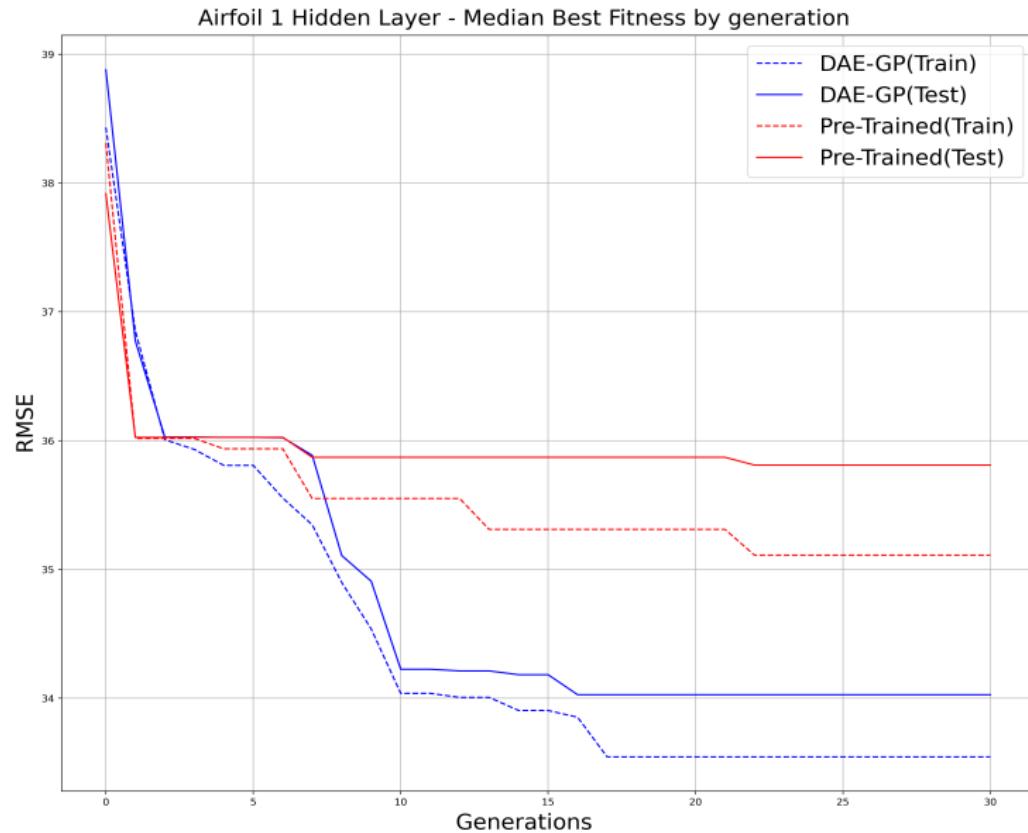
Kontrollexperiment: Reduzierung der DAE-LSTM Dimension für Airfoil Datensatz

Frage: Welchen Einfluss hat die Reduktion der DAE-LSTM auf 1 hidden Layer (HL)?

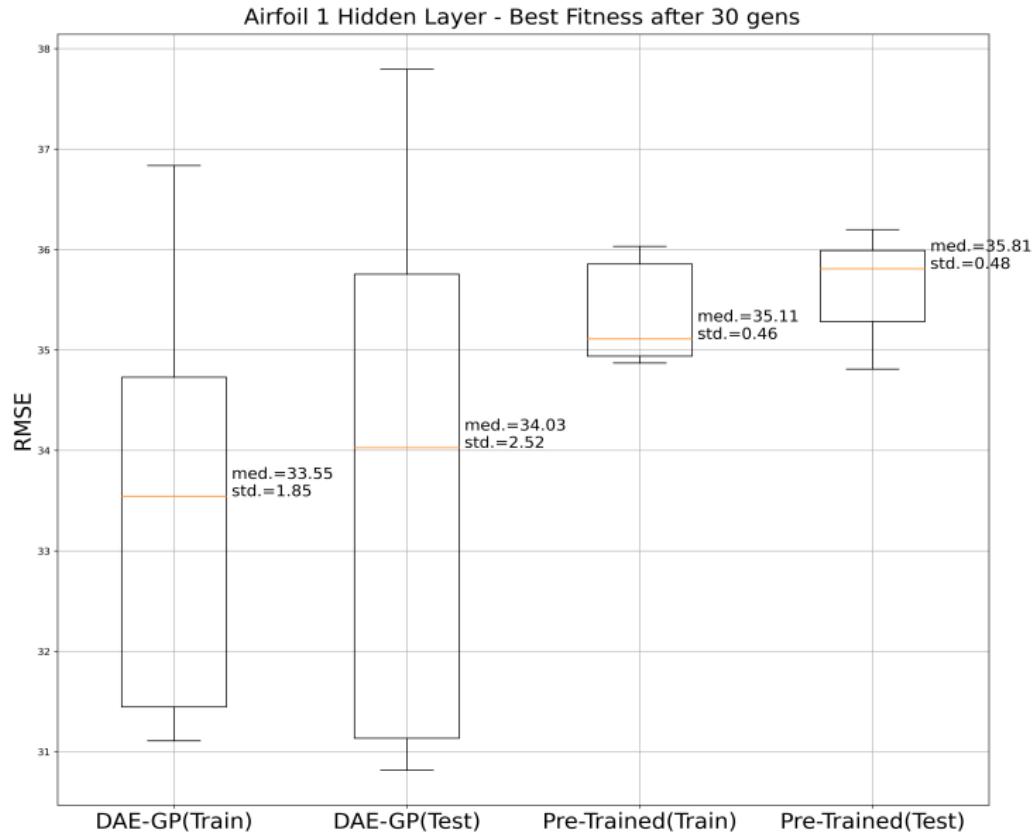
Erwartung: Negativer Einfluss von Pre-Training auf die Güte der gefundenen Lösungen (Fitness/Größe) durch sinkenden Modell Performance der DAE-LSTMs¹³

¹³Erhan u. a. (2009)

Airfoil - Entwicklung der Fitness (1 HL)

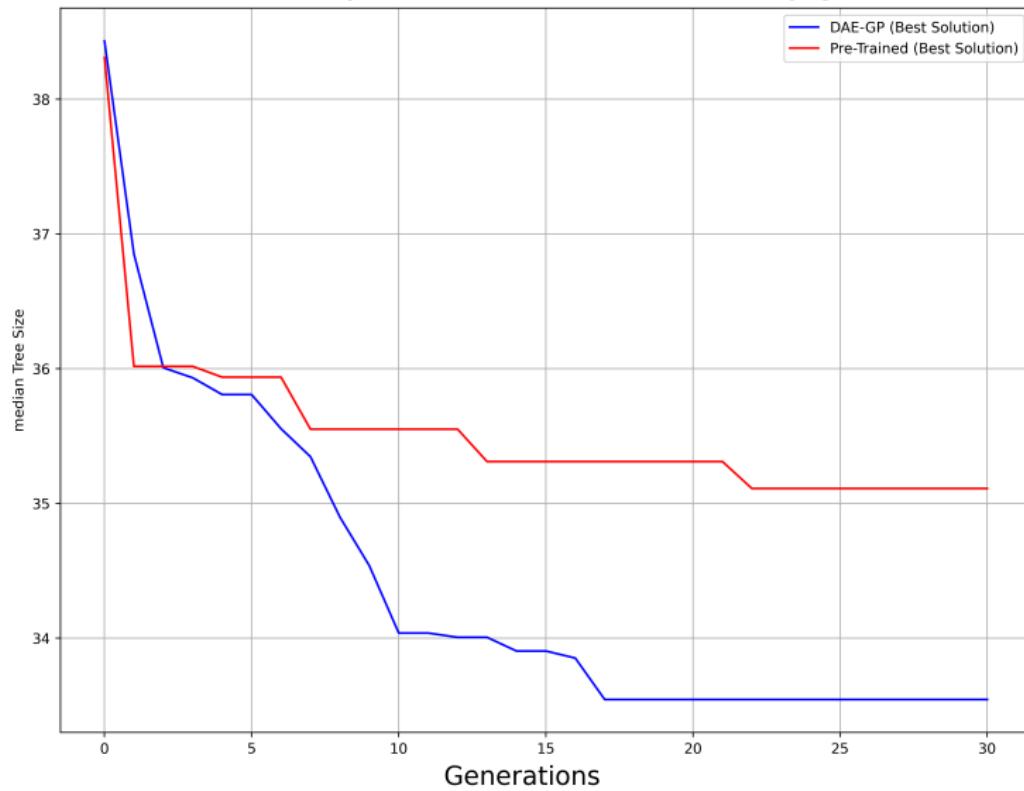


Airfoil - Verteilung der finalen Fitness (1HL)



Airfoil - Größe der besten Lösungskandidaten(1HL)

Airfoil 1 Hidden Layer - median Solution Size by generation



Anwendung auf weitere Datensätzen

Airfoil Datensatz: Erste Ergebnisse deuten bei einer ausreichenden Anzahl von Hidden Layern auf einen positiven Effekt der Pre-Training Strategie hin:

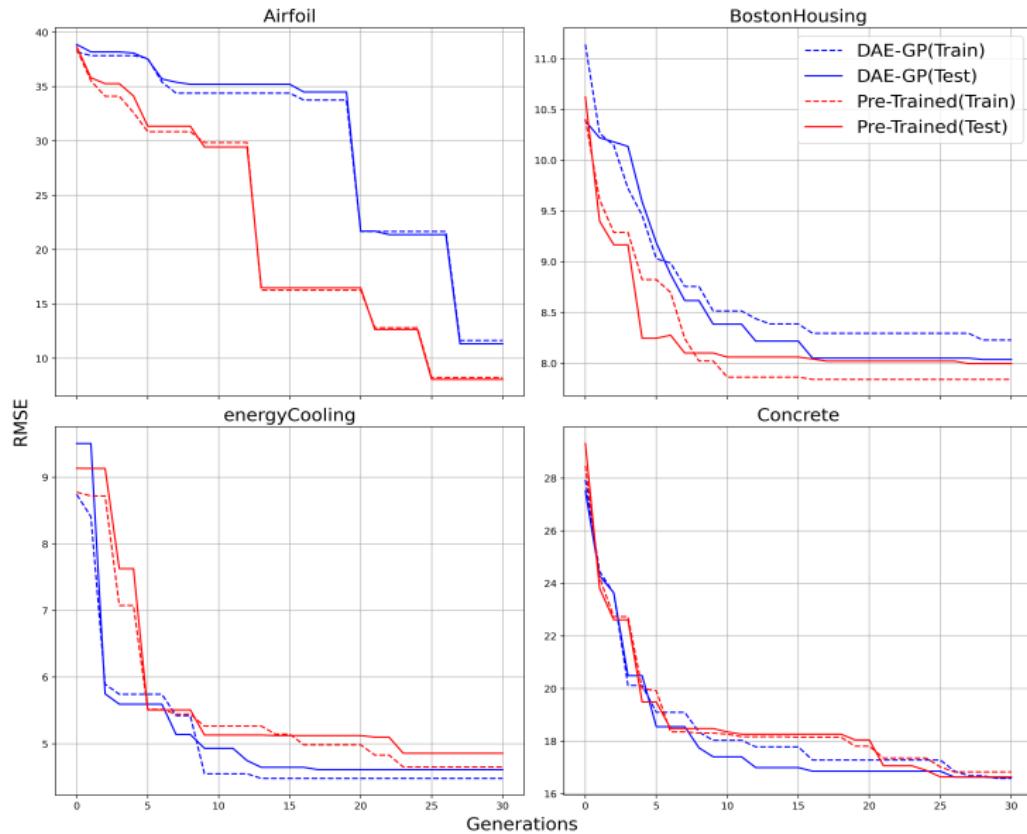
- Höhere Fitness der gefundenen Lösungen
- Kleinere Größe der gefundenen Lösungen

Daher: Ausweitung des Experiments auf weitere Datensätze

Übersicht Datensätze

Problem	Observations	Features
Airfoil	1503	5
Boston_Housing	506	13
Energy_Cooling	768	8
Concrete	1030	8

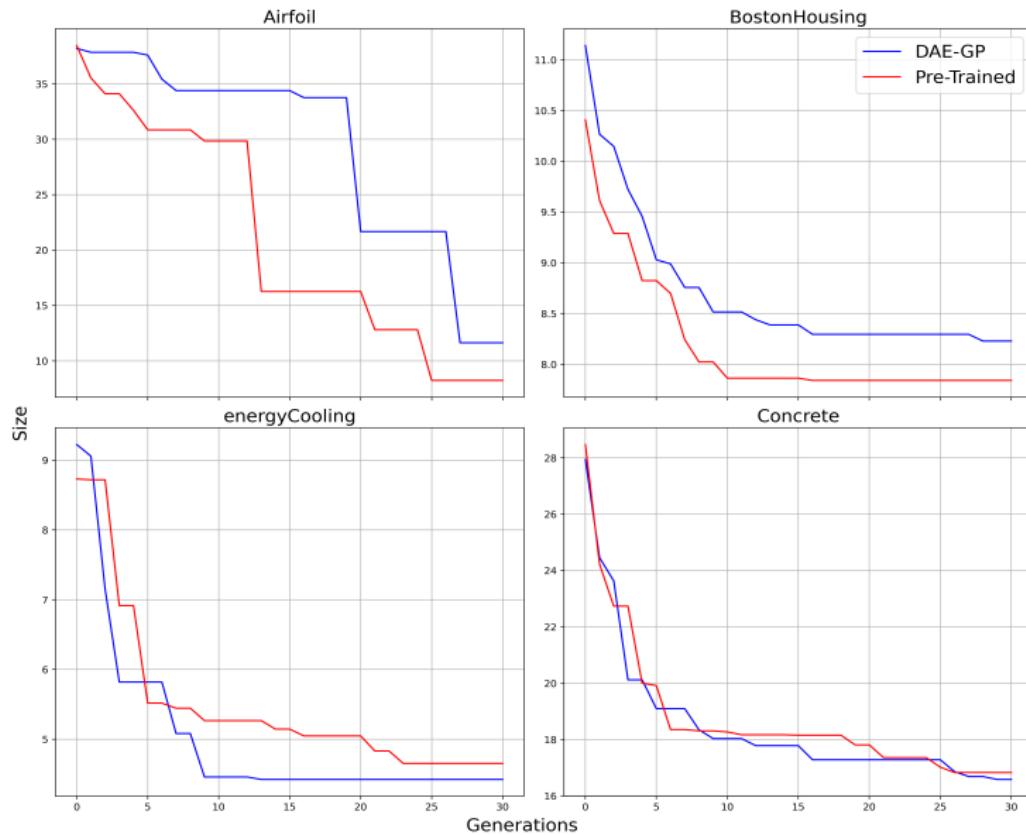
Median Fitness - Symbolische Regression



Fitness (Median) - Symbolische Regression

Problem	Hidden-Layers	Set	DAE-GP	Pre-Trained	P-Value	Cliffs-Delta
Airfoil	1	Train	33.55	35.11	0.02**	0.64
	1	Test	34.03	35.81	0.14	0.40
Airfoil	2	Train	11.64	8.24	0.31	-0.28
	2	Test	11.34	8.06	0.57	-0.16
Boston_Housing	2	Train	8.23	7.84	0.29	-0.29
	2	Test	8.04	8	0.71	-0.11
Energy(Cooling)	2	Train	4.42	4.65	0.02**	0.63
	2	Test	4.61	4.86	0.29	0.29
Concrete	2	Train	16.58	16.83	0.97	0.02
	2	Test	16.62	16.64	0.52	0.18

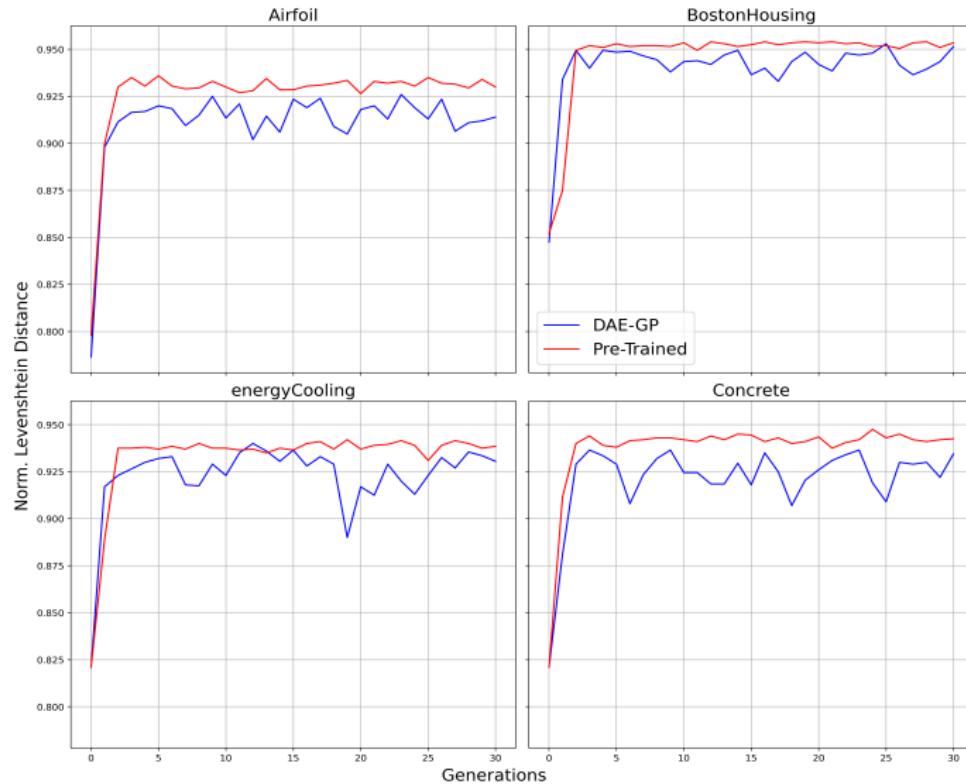
Lösungsgröße (Median) - Symbolische Regression



Auswertung Lösungsgröße (Median) - Symbolische Regression

Problem	Hid.Layers	DAE-GP	Pre-Trained	P-Value	Cliffs-Delta
Airfoil	1	33.55	35.11	0.02**	0.64
Airfoil	2	11.64	8.24	0.31	-0.28
Boston_Housing	2	8.23	7.84	0.29	-0.29
Energy(Cooling)	2	4.42	4.65	0.02**	0.63
Concrete	2	16.58	16.83	0.97	0.02

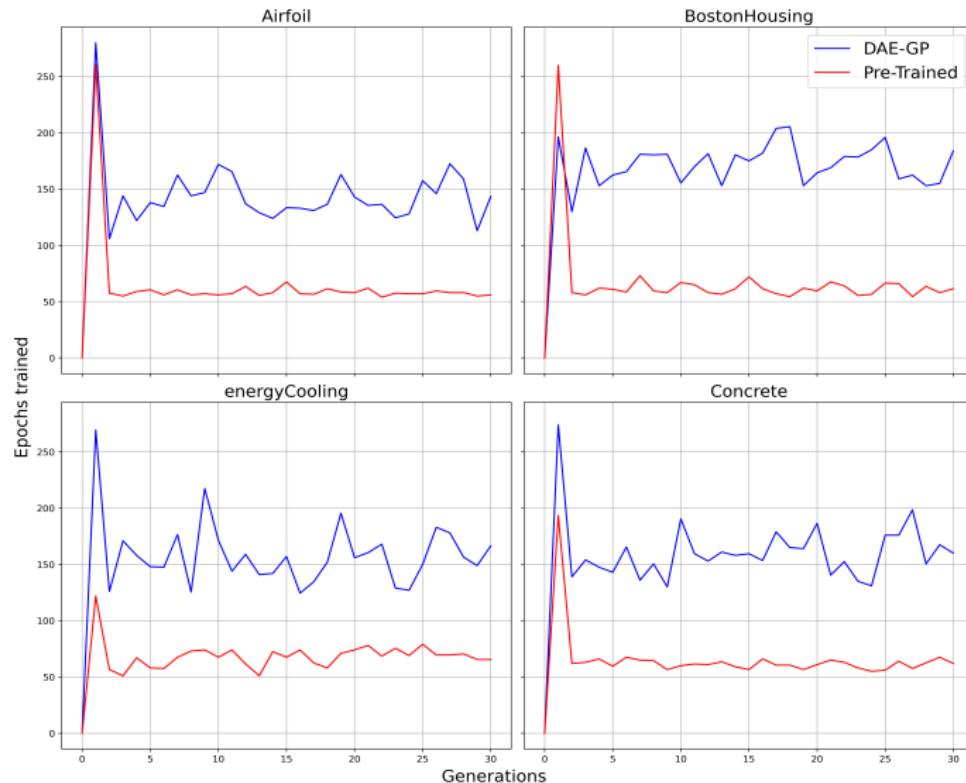
Populationsdiversität (Median) - Symbolische Regression



Auswertung Populationsdiversität (Median) - Symbolische Regression

Problem	Hid. Layers	DAE-GP	Pre-Trained	P-Value	Cliffs-Delta
Airfoil	1	0.34	0.49	0.02**	0.36
Airfoil	2	0.91	0.93	0.00***	0.88
Boston_Housing	2	0.94	0.95	0.00***	0.82
Energy(Cooling)	2	0.93	0.94	0.00***	0.80
Concrete	2	0.93	0.94	0.00***	0.88

Trainingsepochen pro Generation (Median) - Symbolische Regression



Auswertung Trainingsepochen pro Generation (Median) - Symbolische Regression

Problem	Hid.Layers	DAE-GP	Pre-Trained	P-Value	Cliffs-Delta
Airfoil	1	190.93	60	0.00***	-0.94
Airfoil	2	151.87	64.95	0.00***	-0.88
Boston_Housing	2	182.84	67.27	0.00***	-0.88
Energy_Cooling	2	169.39	69.87	0.00***	-0.90
concrete	2	171.69	64.9	0.00***	-0.92

Zusammenfassung- Symbolische Regression

Keine Evidenz für einen statistisch signifikanten Einfluss von Pre-Training auf die erzielte Lösungsqualität oder die erzielte Lösungsgröße in den durchgeföhrten Experimenten!

Vorteile durch Pre-Training:

- ① Signifikante Erhöhung der Populationsdiversität in allen Experimenten
- ② Signifikante Reduktion der Anzahl an Trainingsepochen pro Generationen in allen Experimenten

Section 6

Alternative Pre-Training Strategien

Eingeschobenes Pre-Training in der 2. Generation

Ansatz: Deutliche Fitnessverbesserung finden oft bereits in der ersten Generationen der evolutionären Suche statt

- Sampling des DAE-LSTMs der ersten Generation M_1 zur Initialisierung der Pre-Training Population \hat{P}
- Training des pre-training modells \hat{M} mit \hat{P} nach Abschluss der ersten Generation
- DAE-LSTM Modelle werden ab Generation 2 mit den Parametern von \hat{M} initialisiert

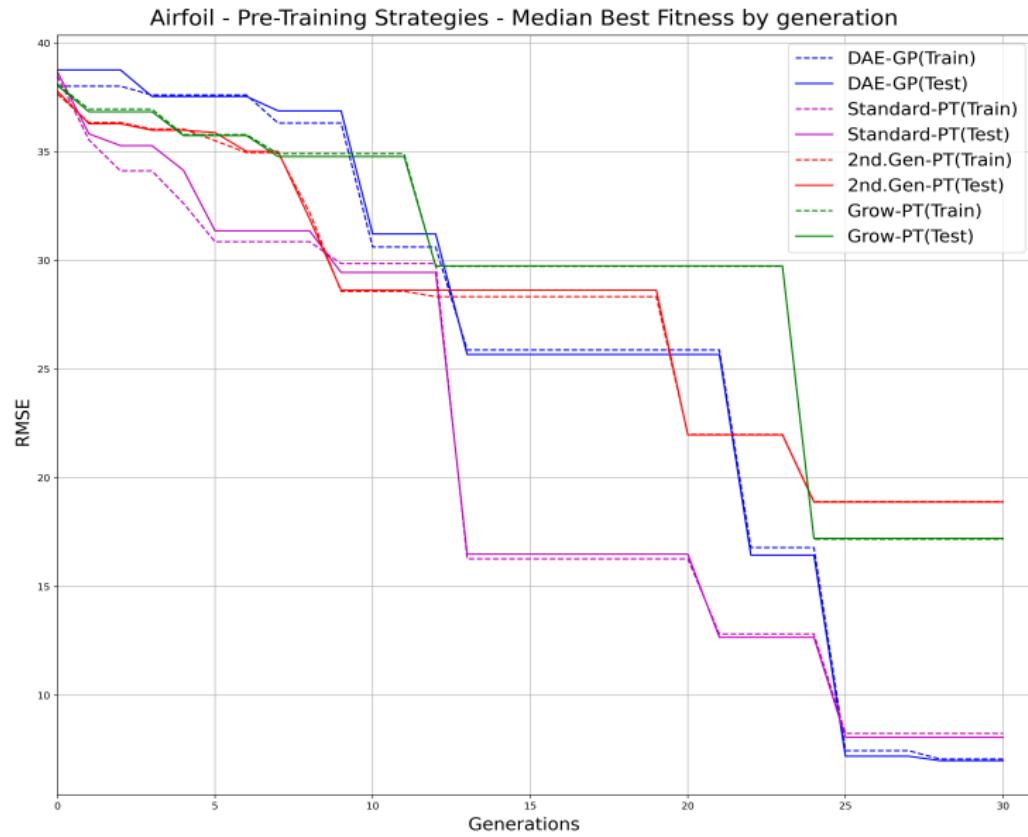
Grow Initialisierung der Pre-Training Population

Ansatz: Initialisierung von \hat{P} ausschließlich mit der Grow Methode¹⁴ anstelle von ramped Half and Half

Hintergrund: Vermutlich geringer Informationsgewinn aus dem Lernen von vollen Lösungsbäume aus \hat{P} mit den gesampleten Populationen der späteren Generationen

¹⁴Koza (1993)

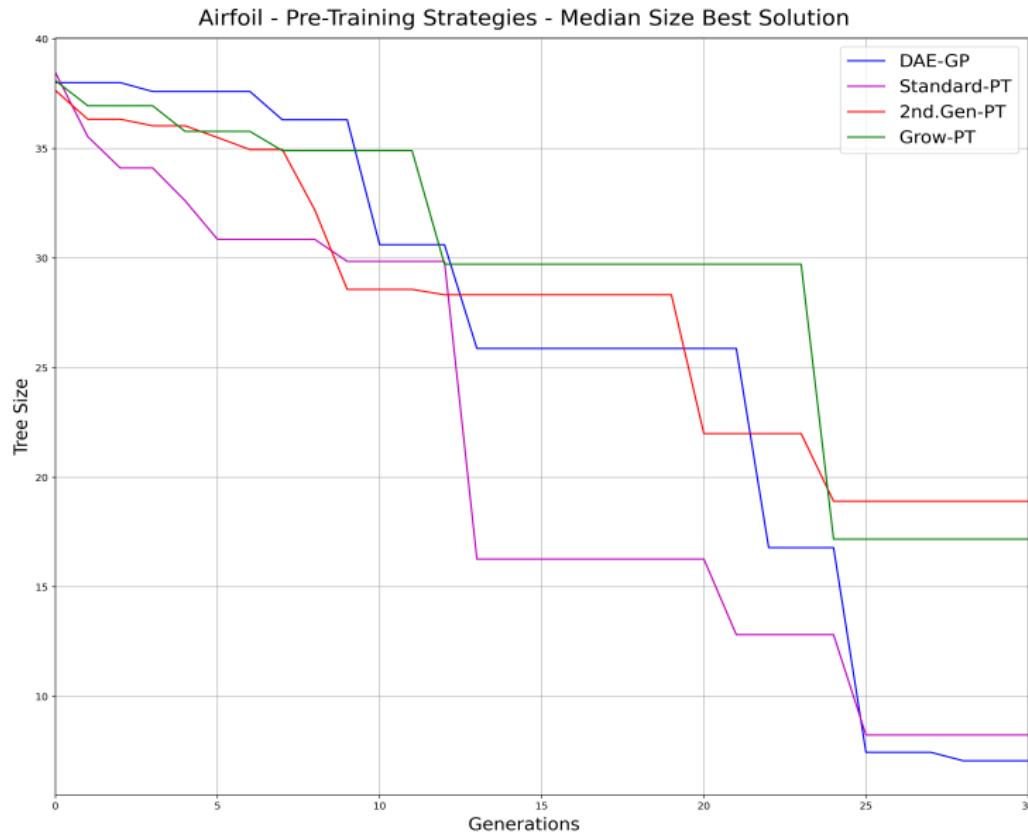
Fitness (Median) - Alternative Ansätze



Auswertung Fitness (Median) - Alternative Ansätze

Set	Standard-PT	2ndGen	P-Val(1)	GrowInit	P-Val(2)
Train	8.24	18.91	0.31	17.18	0.27
Test	8.06	18.88	0.62	17.21	0.57

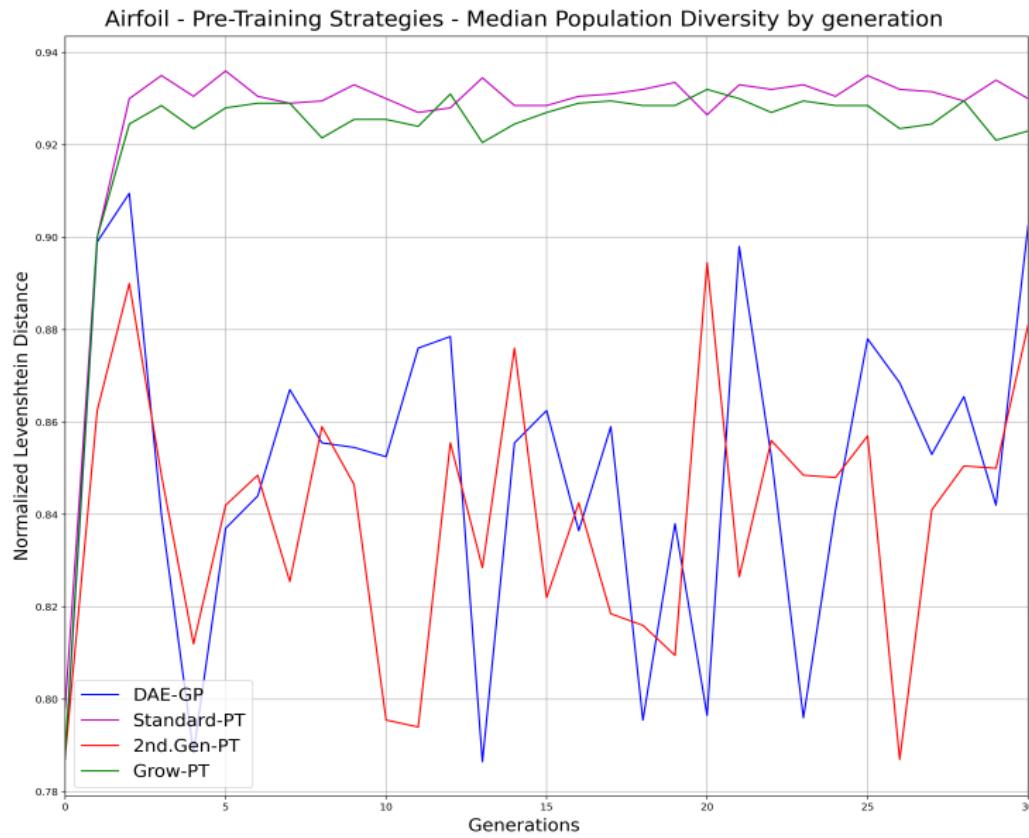
Lösungsgröße (Median) - Alternative Ansätze



Auswertung Lösungsgröße (Median) - Alternative Ansätze

Standard-PT	2ndGen	P-Val(1)	GrowInit	P-Val(2)
8.24	18.91	0.31	17.18	0.27

Populationsdiversität (Median) - Alternative Ansätze

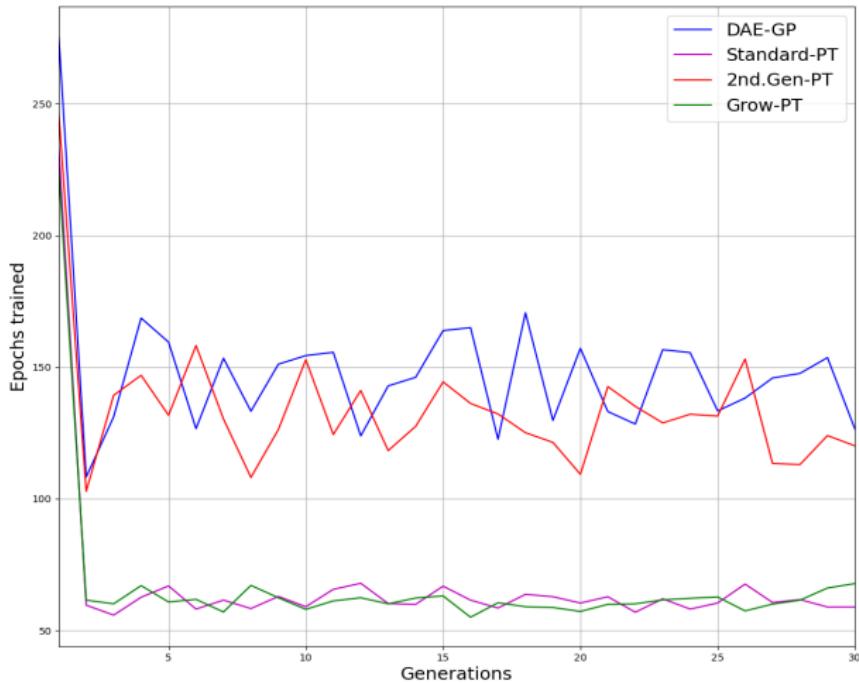


Auswertung Populationsdiversität (Median) - Alternative Ansätze

Standard-PT	2ndGen	P-Val(1)	GrowInit	P-Val(2)
0.93	0.85	0.00***	0.93	0.00***

Trainingsepochen (Median) - Alternative Ansätze

Airfoil - Pre-Training Strategies - Mean Training epochs



Auswertung Trainingsepochen (Median) - Alternative Ansätze

Standard-PT	2ndGen	P-Val(1)	GrowInit	P-Val(2)
57.5	126	0.00***	57	0.31

Section 7

Fazit (vorläufig)

Vorteile

Der Einsatz von Pre-Training dient:

- der Erhöhung der Populationsdiversität (mögliche Strategie zur Kontrolle des Explore/Exploit Verhaltens von DAE-GP?)
- der Reduktion der Trainingsepochen pro Generation (Reduktion der Rechenleistung, insbesondere bei komplexen DAE-LSTM nützlich?)

Nachteile

- DAE-GP verliert durch Einsatz von Pre-Training die Möglichkeit die Anzahl künstlicher Neuronen pro verstecktem Layer anzupassen
- Erhöhung der Rechenzeit/Ressourcen: DAE-LSTMs müssen ausreichend dimensioniert sein (2+ Hidden Layer), aktuelle Publikationen nutzen nur einen einzelnen Hidden Layer^{15 16}.
- Keine signifikanten Vorteile für die Qualität der gefundenen Lösungen

¹⁵Wittenberg und Rothlauf (2022)

¹⁶Wittenberg (2022)

Section 8

References

References I

- Dua, D. und Graff, C. (2017) „UCI Machine Learning Repository“. University of California, Irvine, School of Information; Computer Sciences. Verfügbar unter: <http://archive.ics.uci.edu/ml>.
- Erhan, D. u. a. (2009) „The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training“, in D. van Dyk und M. Welling (Hrsg.) *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR (Proceedings of Machine Learning Research), S. 153–160. Verfügbar unter: <https://proceedings.mlr.press/v5/erhan09a.html>.
- Koza, J.R. (1993) „Genetic programming - on the programming of computers by means of natural selection“, in *Complex Adaptive Systems*.
- Mühlenbein, H. und Paass, G. (1996) „From Recombination of Genes to the Estimation of Distributions I. Binary Parameters.“, in *From Recombination of Genes to the Estimation of Distributions I. Binary Parameters*, S. 178–187.

References II

- Probst, M. und Rothlauf, F. (2020) „Harmless Overfitting: Using Denoising Autoencoders in Estimation of Distribution Algorithms“, *Journal of Machine Learning Research*, 21(78), S. 1–31. Verfügbar unter:
<http://jmlr.org/papers/v21/16-543.html>.
- Rothlauf, F. (2011) *Design of Modern Heuristics: Principles and Application*, Natural Computing Series. Verfügbar unter:
<https://doi.org/10.1007/978-3-540-72962-4>.
- Vincent, P. u. a. (2008) „Extracting and composing robust features with denoising autoencoders“, in *Proceedings of the 25th International Conference on Machine Learning*, S. 1096–1103. Verfügbar unter:
<https://doi.org/10.1145/1390156.1390294>.
- Wittenberg, D. (2022) „Using Denoising Autoencoder Genetic Programming to Control Exploration and Exploitation in Search“, in E. Medvet, G. Pappa, und B. Xue (Hrsg.) *Genetic Programming*. Cham: Springer International Publishing, S. 102–117.

References III

- Wittenberg, D. und Rothlauf, F. (2022) „Denoising Autoencoder Genetic Programming for Real-World Symbolic Regression“, in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. New York, NY, USA: Association for Computing Machinery (GECCO '22), S. 612–614. Verfügbar unter: <https://doi.org/10.1145/3520304.3528921>.
- Wittenberg, D., Rothlauf, F. und Schweim, D. (2020) „DAE-GP: Denoising Autoencoder LSTM Networks as Probabilistic Models in Estimation of Distribution Genetic Programming“, in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. New York, NY, USA: Association for Computing Machinery (GECCO '20), S. 1037–1045. Verfügbar unter: <https://doi.org/10.1145/3377930.3390180>.